

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開平11-184678

(43)公開日 平成11年(1999) 7月 9 日

(51)Int.Cl.⁶

識別記号

F I

G 0 6 F 7/58

G 0 6 F 7/58

B

審査請求 未請求 請求項の数10 O L (全 12 頁)

(21)出願番号 特願平9-358015

(22)出願日 平成9年(1997)12月25日

(71)出願人 000003078

株式会社東芝

神奈川県川崎市幸区堀川町72番地

(72)発明者 浜崎 俊治

東京都青梅市末広町2丁目9番地 株式会
社東芝青梅工場内

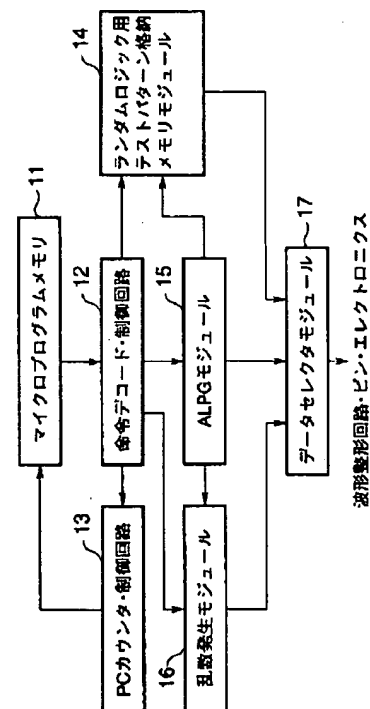
(74)代理人 弁理士 鈴江 武彦 (外6名)

(54)【発明の名称】 パターン発生器

(57)【要約】

【課題】本発明は、従来のパターン発生器に乱数発生モジュール（高速プログラマブルパターン発生装置）を追加して設けることにより、従来の機能を損なうことなく従来の装置を有効に活用して、かつ高度なプログラミング知識を一切必要とせずに、従来のパターン発生器では達成し得ない検出率の高いランダムなテストパターン列が生成できるパターン発生器を提供することを課題とする。

【解決手段】マイクロプログラムメモリに格納されたプログラムの動作シーケンスに従いメモリIC試験検査用のテストパターンを生成するパターン発生器に、上記プログラムの制御の下に高速のプログラマブルパターンを発生する乱数発生モジュール16と、当該モジュールで発生したランダムロジック用のテストパターンデータを格納するメモリモジュール14を追加して設け、更にALPGモジュール15との同時使用を可能にしたことを特徴とする。



【特許請求の範囲】

【請求項1】 マイクロプログラムメモリに貯えられたプログラムに従う動作シーケンスにより半導体メモリ等の試験検査用のランダムテストパターンを生成するパターン発生器に於いて、

上記プログラム制御の下にプログラマブルテストパターンを発生する乱数発生モジュールと、

前記乱数発生モジュールで発生したテストパターンデータを格納するメモリモジュールとを具備してなることを特徴とするパターン発生器。

【請求項2】 マイクロプログラムメモリに貯えられたプログラムに従う動作シーケンスによりテストパターンを生成する、アルゴリズムックパターン発生手段を備えたパターン発生器に於いて、マイクロプログラムメモリに乱数モジュールの制御フィールドを追加するとともに、命令デコードのデコードの数を増加して、半導体メモリ等の試験検査用のランダムテストパターンを生成できるようにした請求項1記載のパターン発生器。

【請求項3】 乱数発生モジュールは「Nパターン」のパターン発生回数で、ランダムアドレスとデータを発生する請求項1又は2記載のパターン発生器。

【請求項4】 乱数発生モジュールは、乱数系列による、アドレスパターンとデータパターンの生成を同時に実行処理する処理手段をもつ請求項1又は2又は3記載のパターン発生器。

【請求項5】 乱数発生モジュールによる乱数系列の発生は、テストシステム側のプログラム制御が可能で、初期値をプログラミングすることにより自動生成される請求項4記載のパターン発生器。

【請求項6】 乱数生成値の同値検出機能を備え、異なる値のみを出力可能とした請求項1又は2又は3又は4記載のパターン発生器。

【請求項7】 乱数発生モジュールは、生成されるアドレス値及びデータ値の乱数が異なる値となるまで同一値をスキップする機能をもつ請求項1又は2又は3又は4又は6記載パターン発生器。

【請求項8】 乱数発生モジュールは、M系列発生回路を複数個使用して多ビット出力回路を構成し、初期値を出力した後、自動的に乱数値を生成する機能を有してなる請求項1又は2又は3又は4又は6又は7記載パターン発生器。

【請求項9】 テストパターンをリアルタイムに発生する機能と、予め生成した値を格納しておき、その値をテストパターンとする機能を有してなる請求項1又は2又は3又は4又は6又は7記載パターン発生器。

【請求項10】 パーソナルコンピュータの拡張スロットに挿入される拡張用ボードに実装される請求項1又は2又は3又は4又は6又は7記載パターン発生器。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、例えば半導体テストシステム、ボードテストシステム、モジュール部品試験検査装置、計測機器等の各種電子機器及び部品類に適用されるパターン発生器に関する。

【0002】また本発明は、例えばフラッシュメモリ等、大容量で高速アクセスが可能な半導体メモリ装置のテストシステムに適用して好適なパターン発生器に関する。また本発明は、例えばパーソナルコンピュータ等の拡張スロットに挿入される拡張ボードに実装可能なパターン発生器に関する。

【0003】

【従来の技術】半導体テストシステム、ボードテストシステム等の試験検査に適用されるパターン発生器は、試験するDUT (Device Under Test) に印加する情報と、DUTから出力される期待値及びDUTに印加する信号の方向とがそれぞれテストパターンとして予め格納されており、主にDUTの機能試験を行なう際に用いられる。

【0004】このパターン発生器で扱えるテストパターンは、一般ディジタルICやCPU等のランダムロジックIC用のものと、アドレス関数やデータ関数を効率よく発生することのできるメモリIC用のものとに大別できる。前者はランダムロジックパターン発生器と称され、後者はアルゴリズムックパターン発生器と称される。更にこの2つの機能を搭載したパターン発生器も存在する。両者の共通点は、マイクロプログラム制御方式によりコントロールされ、柔軟なテストパターンが発生できるようになっている点である。

【0005】図10はディジタル半導体テストシステムの代表的な構成例を示している。この図10に示すテストシステムに於いて、パターン発生器3の機能は前述の通りである。タイミング発生器2は時間的な情報を定義するモジュールであり、当該タイミング発生器2、及びパターン発生器3の各情報は波形整形回路5に印加される。

【0006】波形整形回路5は、パターン発生器3とタイミング発生器2の情報をもとに、DUTに印加される波形フォーマットを作る回路である。波形フォーマットの種類としては、RZ (Return to Zero)、NRZ (Non Return to Zero)、EX-OR (Exclusive OR) 等がある。

【0007】上記波形整形回路5により整形、整時された波形は、ピンエレクトロニクスのドライバ6aに入力されて、プログラムされたハイ/ローレベルにクランプされた後、一定インピーダンス(50Ω)の信号線を介し入力波形としてDUT8に印加される。

【0008】一方、DUT8からの出力は、アナログコンパレータ6bで出力レベルをDAC6eの基準値(スレッショルドレベル)と比較された後、比較レジスタ4に転送される。

【0009】比較レジスタ4では、タイミング発生器2の時間情報（ストロブ）とパターン発生器3より入力した期待値パターンとを比較して機能的な合否の判定を行なう。

【0010】パラメトリック測定ユニット7は、電圧や電流を測定するモジュールである。ピンエレクトロニクス部6には信号を分岐するリレー6dが設けられている。上記パターン発生器3を含む各機能モジュールはすべてコントローラインタフェース1及びシステムバス9を介してプログラムコントロールされる。

【0011】図11は従来のパターン発生器の構成を示すブロック図である。ここでは、ランダムロジック用パターン発生器（ランダムロジックパターン発生器）とメモリ用パターン発生器（アルゴリズムパターン発生器）の双方を含んだ構成を示している。

【0012】この図11に示すような構成のパターン発生器は、一般にランダムロジックパターン発生器3Aが装置全体をコントロールする。図11に於いて、3a乃至3iはランダムロジックパターン発生器3Aの構成要素をなすもので、3aはプログラムカウンタ（PC）であり、マイクロプログラムメモリ3bをアクセスするためのカウンタである。マイクロプログラムメモリ3bは、パターン発生器3の動作シーケンスを決める命令や、アルゴリズムパターン発生器3Bのデータ・アドレス演算制御ユニット3p、及び制御データ発生部3qのデータを格納している。このマイクロプログラムメモリ3bに種々の命令を格納し動作シーケンスをプログラムすることにより、目的のテストパターンを発生させることができる。

【0013】命令デコード3cはマイクロプログラムメモリ3bの命令フィールドに格納されているデータをデコードして、命令制御ロジック3dの回路を選択する。パターン発生器3の命令の種類は数種類で、命令の数は40個以下の場合が殆どである。

【0014】これは、より高速に命令処理しなければならないことと、パターン発生シーケンスのみに注目すると機能的に多くないことを意味している。ループカウンタ3e、及びサブルーチンスタック3fは、同一パターンの発生を無駄なく（効率よく）生成させるためのレジスタである。これらのレジスタは通常2乃至16個程度有しており、多重ループや多重ネestingが可能になっている。

【0015】PC制御ロジック3gは、次に実行する命令をコントロールして、その値をプログラムカウンタ3aに転送する。また、パターン発生器3の起動（スタート）、停止（ストップ）、一時停止（ポーズ）、タイマ割り込み、マッチ割り込み等のフラッグをもとにプログラムカウンタ3aをコントロールする。

【0016】アドレス発生回路3hは、メモリ群3iをアクセスするアドレスを発生するための回路である。メ

モリ群3iは、試験デバイス（DUT8）に印加されるテストパターンや、ピンエレクトロニクスのI/O制御情報、波形整形情報（ウェーブフォーマット）等の各情報を記憶する大容量メモリ群である。これらのメモリからの出力データが、試験デバイス（DUT8）に実際に印加されるデータ（テストパターンデータ）となる。

【0017】以上が、ランダムロジックパターン発生器3Aの構成及び機能である。一方、アルゴリズムパターン発生器（以下ALPGモジュールと称す）3Bは、マイクロプログラムメモリ3bのALPG制御フィールドの情報と、命令制御ロジック3dから入力される情報をもとに動作を決定する。

【0018】ALPGモジュール3Bは、主にメモリIC等のテストパターンの発生を目的に実現されたパターン発生器であり、演算処理した結果を出力する方法が簡単で、テストデータを格納するメモリを必要としないため、低コストで実現できる。

【0019】ALPGモジュール3Bのデータ・アドレス演算制御ユニット3pは、メモリデバイスに印加するデータパターンやアドレスパターンを演算する回路である。演算式を決める情報は、マイクロプログラムメモリ3bのALPG制御フィールドから与えられ、演算の繰り返し回数、演算結果の最大値、最小値等の情報は、命令制御ロジック3dから与えられる。

【0020】ALPGモジュール3Bの制御データ発生部3qは、試験メモリへのライト／リード信号制御やI/O制御を行う。また、アドレス発生-2回路3sで発生されたアドレス情報を、メモリ群3iをアクセス可能にするコントロールも行なう。

【0021】以上がアルゴリズムパターン発生器（ALPG）3Bの構成及び機能である。上記したように、従来のテストシステムに搭載されているパターン発生器は、プログラミング技術に追うところが多く、プログラミング方法によっては、複雑なテストパターンを作成することが可能である。

【0022】

【発明が解決しようとする課題】しかしながら、上記したパターン発生器に於いては、試験デバイスに入力テストパターンをプログラミングすると同時に、試験デバイスからの応答パターン、つまり期待値パターンも同時にテストパターンとしてプログラミングする必要がある。従ってマイクロプロセッサのようなランダムロジックLSI等のテストパターンに対しては、論理シミュレーションの力を借りないとテストパターンの作成が困難であった。また、例えばフラッシュメモリ等、リード／ライトのアクセスタイムが異なるようなメモリデバイス等の高速テストパターンの発生に対しても、その全ての高速入出力データを記憶する手段が必要となり、特に容量の大きなこの種高速メモリデバイスのテストに対して複数種（Nパターン）のランダムパターン発生によるテスト

5

機能を実現させようとする、高速アクセスが可能な膨大な容量のメモリ装置が必要となり、システム構成が著しく複雑となることから実現性に乏しいという問題があった。

【0023】本発明は上記実情に鑑みなされたもので、従来の装置を有効に活用して、高度なプログラミング知識を一切必要とせずに、簡単かつ安価な構成で、従来のパターン発生器では達成し得ない検出率の高いランダムなテストパターン列が生成でき、効率の良いテストパターンの発生が期待できるパターン発生器を提供することを目的とする。

【0024】

【課題を解決するための手段】本発明は、概略的には従来機能のパターン発生器に乱数発生モジュールを追加することにより、従来のパターン発生器では不可能なテストパターン列を生成できるようにした。しかも乱数生成部がモジュール化されることから、高度なプログラミング知識を必要としない。また、ALPGと同時起動が可能であるため、テストパターンの作成が増え、非常に効率の良いテストパターンの発生が期待できる。

【0025】上記した現状のテストシステムに搭載されているパターン発生器は、プログラミング技術に追うところが多く、プログラミング方法によっては、複雑なテストパターンを作成することが可能である。しかし、試験デバイスに入力テストパターンをプログラミングすると同時に試験デバイスからの応答パターン、つまり期待値パターンも同時にテストパターンとしてプログラミングする必要がある。

【0026】マイクロプロセッサのようなランダムロジックLSIなどのテストパターンは、論理シミュレーションの力を借りないとテストパターンの作成が困難な状態になっているが、別の見方をすれば、テストパターンの作成方法や作成するためのツールがあるということである。

【0027】本発明は、ランダムロジックLSI用のテストパターン作成ではなく、メモリIC用のテストパターン生成を対象になされたものである。一般的にメモリ用のテストパターンは、ALPGの演算回路を如何に制御して、テストパターンを生成するかにかかっている。テストパターン生成の種類としては、試験メモリをアクセスするアドレス関数の生成、メモリセルをリード/ライトするデータ関数の生成と、ライト/リードする切り替えパターンのみである。このうち、アドレス関数は、簡単なある数列手順（加算するか、減算するか、何倍するか、何の値で割るか等）に従って発生しており、どのような値になったら、演算結果値を初期化するか、補数をとるか、最大値あるいは最小値に達したらどのように処理するか等をプログラミングすることになる。

【0028】一方、テストパターン関数は、アドレス関数同様の演算も可能であるが、一般的には、初期値を与

6

えておき、その値の補数を繰り返しとることにより、データパターンを生成するような手法がとられる。もっとも、アドレス関数はメモリをアクセスするアドレス値を出したままでもよいが、データパターンはリード時に期待値と比較する関数上、メモリのアドレスが決定されれば、書き込んだときのデータ値と同じ値を生成できる演算（プログラミング）を行う必要がある。

【0029】メモリのアドレス関数の種類を大きく分けると、「Nパターン」、「 $N3/2$ 乗パターン」、「 $N2$ 乗パターン」（但しNをアドレス指定ビット数としたとき）等がある。最近の大容量メモリでは、「Nパターン」以外はテスト時間の関係上、検出率が非常に高いにも関わらず、実用的ではない。従って「Nパターン」で如何に効率よく、しかも検出率を上げるかがアドレス関数、データ関数生成のプログラミングに要求される。

【0030】この際の代表的なアドレス関数によるパターン発生数の計算式とサイクルタイムが100nsのランダムアクセスメモリを想定したときのテスト時間を比較した例を図9に示している。

【0031】本発明は、「Nパターン」のパターン発生回数で、ランダムアドレスとデータを発生するモジュールを提供することを目的とする。このモジュールは、本来、メモリのアクセスが、ランダムアクセスであることと、書かれるデータも、ランダム値を持ったデータであることに起因している。

【0032】

【発明の実施の形態】本発明は、上述したように、従来のパターン発生器に「乱数発生モジュール」と「発生したデータを格納するメモリモジュール」を追加することにより実現される。このパターン発生器の開発にあたり、アプリケーションの面から次の仕様を満足することにした。

【0033】（1）. 従来のパターン発生器の機能は犠牲にしないものとする。

従来の、パターン発生器の機能モジュールに乱数発生のためのモジュールを追加することにより実現する。このモジュールの制御は、マイクロプログラムメモリに乱数モジュールの制御フィールドを追加するとともに、命令デコードのデコードの数を増加することで対応する。

【0034】（2）. 乱数系列の発生は、テストシステム側のプログラム制御が可能であることと乱数発生そのもののプログラミングは行なわないで、初期値（乱数に種を与えるのみ）をプログラミングすることにより自動生成すること。

【0035】プログラミングを簡単にするため、乱数列そのものの発生に関してはユーザプログラミングする必要はなく、初期値のみ与えることにより、異なった乱数列を自動的に生成する方式とする。

【0036】（3）. アドレス及び、データ値の乱数は、同一値をスキップ（破棄する）して、新たな値を生

成させる機能を持つこと。

通常の乱数の値をそのまま採用すると、必ず同一の値が発生する。例えば、この値をメモリのアドレッシングに使用した場合、同じメモリセルをアクセスすることになり、効率の良いアクセスとはいえない。また、すべてのメモリセルが最低一度はアクセスされるという保証も無いことになる。従って、異なった値が生成されるまで、スキップする機能を実現する。

【0037】(4)．乱数の値の発生方法は、リアルタイム(実時間)に発生する機能と予め発生された値を格納しておき、その値をテストパターンとする機能を有すること。

【0038】乱数系列をリアルタイムで発生するとき、その値を一時的に格納することなく、テストパターンとして、試験デバイスに印加することができるため、回路は簡単になるが、すべてのメモリセルが選択される保証がない。同一値の異なった値が生成されるまでスキップすればよいが、100nsで1個のテストパターンを実行しなければならないパターン発生器では、時間的に不可能である。従って、リアルタイムにパターンを発生する機能では、メモリのデータパターンの値のみとし、アドレッシングを目的には使用しないことを想定している。

【0039】パターン発生の実行開始前に乱数の値を予め格納しておく方法は、同一値の場合、異なった値が生成されるまでスキップすることが可能なので、無駄のないテストパターンの実行ができる反面、格納するメモリが必要になる。この機能は試験メモリのアドレッシングにも使用できる。

【0040】以下図面を参照して本発明の一実施形態を説明する。図1は本発明の実施形態によるパターン発生器の構成を示すブロック図である。この図1に示すパターン発生器は、例えばパーソナルコンピュータ等の拡張スロットに挿入して用いられる、例えば半導体テストシステム、ボード・モジュール部品テストシステム等の各種試験検査装置用拡張ボードとして提供できる。

【0041】図1に於いて、11は、図示しないCPUにより実行されるパターン発生器の制御プログラム等を格納するマイクロプログラムメモリであり、12はマイクロプログラムメモリ11に貯えられたマイクロプログラムを解釈し制御信号を生成する命令デコード・制御回路である。

【0042】13はマイクロプログラムメモリ11の内容を更新制御するPCカウンタ・制御回路である。14は後述するアルゴリズムミックパターン発生器15で生成されたランダムロジック用テストパターンを格納するメモリモジュールである。15はマイクロプログラム制御の下に(マイクロプログラムメモリ11のALPG制御フィールド及び制御フィールドの情報をもとに)メモリIC等のテストパターンデータ生成の動作が決定される

アルゴリズムミックパターン発生器であり、ここではALPGモジュールと称す。16は従来では発生不可能であった多岐に亘る効率のよいテストパターンの発生を可能にする、本発明で対象とする乱数発生モジュールであり、その具体的な構成及び機能は後述する。17は上記メモリモジュール14、アルゴリズムミックパターン発生器15、乱数発生モジュール16等で生成されたテストパターンを選択し出力するデータセクタモジュールである。

【0043】上記図1に示す構成のパターン発生器に於いて、マイクロプログラムメモリ11は、パターン発生器を制御するプログラムを格納する。このプログラムメモリ11は、ここでは1Kワード×96ビット構成とし、図2のようにビット割付を行なっている。

【0044】命令デコード・制御回路12は、パターン発生器のシーケンスをコントロールする。PCカウンタ・制御回路13は、命令デコード・制御回路12で命令デコードされた結果により次の動作を決めマイクロプログラムメモリ11をアクセスする。

【0045】メモリモジュール14は、ランダムロジック用のテストパターンデータを格納する。ここでは64Kワード×テストピン数のメモリ構成としている。アルゴリズムミックパターン発生器15は、メモリIC用のテストパターンを生成する。ここではALPGモジュールと称す。このALPGモジュール15は、ここではX、Yのアドレス(12ビット)生成演算ユニットをもっており、16Mのアドレス空間まで対応している。また、試験メモリデバイスへの生成とリード/ライト等のコントロール信号も生成できるようになっている。

【0046】上記ALPGモジュール15で生成されるアドレス情報は、メモリモジュール14や乱数発生モジュール16に搭載されている、データ・ストレージ・メモリを直接アクセスすることも可能である。

【0047】乱数発生モジュール16は、本発明の主要構成要素をなすもので、その内部の構成は図5を参照して後で詳細に説明する。データセクタモジュール17は、マイクロプログラムメモリ11の乱数発生制御フィールドの値により制御されて、メモリモジュール14、アルゴリズムミックパターン発生器15、乱数発生モジュール16等で生成されたテストパターンを選択して出力する。

【0048】図2は上記マイクロプログラムメモリ11の1ワードの構成を示したものである。ここではメモリ構成を1Kワード×96ビットとしている。図中、21は命令フィールドであり、6ビットを使用している。従って命令の種類は最大64種類まで作ることができる。

【0049】22はオペランドフィールドであり、24ビットを使用している。このフィールドの使用目的は、各モジュールのレジスタに設定する値を記述したり、分岐命令では、分岐先アドレスを設定するためのものであ

る。

【0050】23はI/O制御フィールドであり、ピンエレクトロニクスのドライバをイネーブル/ディセーブルするドライバ・イネーブル・メモリのアドレスを指定するフィールドである。ここでは4ビットが用いられ、16種類のドライバイネーブル情報が選択できるようにしている。

【0051】24はコンパレータ制御フィールドであり、試験デバイス(DUT)から出力される値とパターン発生器から出力される期待値パターンとを比較するかどうかを決めるコンパレータ・イネーブル・メモリをアクセスするアドレス情報を記述するフィールドである。ここでは6ビットが用いられ、64種類のコンパレータイネーブル情報が選択できるようにしている。

【0052】25は、RTTC(リアル・タイム・タイミング・コントロール)関連制御フィールドであり、ここでは8ビットが用いられ、主にタイミング発生器のための制御に使用される。

【0053】26はALPGモジュール15を制御するためのALPG制御フィールドである。この実施形態に於けるシステムのALPGモジュール15は、X、Y2次元のアドレスが発生可能なようになっている。両者の演算ユニット及び、関連するレジスタは12ビットになっているため、最大16Mまでのアドレス空間を有するメモリデバイスの試験が可能である。

【0054】27は乱数発生モジュール16を制御するための乱数発生制御フィールドである。ここでは16ビットを使用しており、制御フィールド28に示すように、全く同一機能を有する乱数発生回路を2台搭載している。これは、アドレス用とデータ用の乱数系列を同時に発生可能とするためである。

【0055】このうち、乱数発生Aフィールドは、アドレスパターンを生成するために使用し、乱数発生Dフィールドは、データパターンを発生するために使用することを想定している。しかし、乱数発生Aフィールドをデータパターンのために使用することも可能である。この場合、アドレスパターンは、ALPGモジュール15から発生させる方法が採られる。つまり、このパターン発生器は、ALPGモジュール15と乱数発生モジュール16を同時に実行させることも可能である。

【0056】制御フィールド28について詳細に説明すると、8ビットのプログラムフィールドのうち、下位3ビットは、乱数モジュールの各レジスタに値を設定するコマンドと発生した値を出力するコマンドからなっている。

【0057】この際のコマンドの種類は、

「0」(“000”)→データ出力

「1」(“001”)→初期値レジスタ(図3符号31参照)に初期値を設定する。設定する初期値の値は上記オペランドフィールド22に設定したい値を記述する。

このレジスタの値が乱数系列を決める種(シード)になる。

【0058】「2」(“010”)→最小値レジスタ(図3符号32参照)に発生する値の最小値を設定する。設定方法は上記と同様である。

「3」(“011”)→最大値レジスタ(図3符号33参照)に発生する値の最大値を設定する。

【0059】つまり、乱数系列の値の範囲は、最小値レジスタの値≦発生値≦最大レジスタの値の範囲になる。

「4」(“100”)→発生個数レジスタ(図3符号34参照)に乱数の個数を設定する。

【0060】上記した各レジスタは、それぞれ24ビット構成であり、数値的には、「0」～「16,777,215」までの範囲になる。しかし、試験するメモリデバイスは、種々のビット構成をしているため、このようなビット構成のレジスタが必要になる。

【0061】「5」～「7」(“101”～“111”)→未使用コマンド(拡張用)である。

制御フィールド28の3ビット目は、同値検出回路(図3符号37参照)をイネーブル/ディセーブルするための切り替えビットである。

【0062】同値検出回路(図3符号37参照)をイネーブルすると、一度、生成された値と同じ値が発生したときにスキップ(破棄して無効にする)処理を行ない、一度も発生していない値が生成されるまで繰り返す。しかし、この方法は、乱数列の生成が進めば進むほど、過去に生成した値と異なる確率が段々に減少することになり、最悪の場合、いつまでたっても、生成できない現象が発生する可能性もある。このような処理を行なうときには、タイマ割り込みを起動しておき一定時間内に終了しなかったら、パターンの発生を中止するようにプログラミングすることも必要である。

【0063】制御フィールド28の4ビット目は、データ・ストレージ・メモリ(図3符号39参照)に格納することなく、生成した値(パターン)を垂れ流しにすることを指定する制御ビットである。このビットが有効なときは、同値検出回路(図3符号37参照)は強制的にディセーブルされる。つまり、3ビット目のプログラミングは無視される。

【0064】制御フィールド28の5ビット目は、乱数を格納するデータ・ストレージ・メモリ39のアドレッシングを切り替えるためのビットである。通常は乱数発生モジュール16内で自動的にアドレッシングされるが、このビットをセットすることにより、ALPGモジュール15からのアドレッシングを可能にする。

【0065】制御フィールド28の上位2ビットは、システム使用のため、ユーザプログラミングしても無視される。上記した乱数発生モジュール16の1台分の構成を図6に示す。

【0066】図6に於いて、初期値レジスタ31は、乱

11

数系列の初期値（シード）を設定する。このレジスタの値そのものは、マイクロプログラムメモリ11のオペランドフィールド22から転送される。

【0067】この値は乱数発生・補正回路36で決まった法則（固定的手法）で処理され、確実に乱数が生成されるように補正が加えられる。デフォルトの値は、“FFFFFF”H（16進数）である。

【0068】最小値レジスタ32は、乱数の値の最小値を設定する。デフォルトの値は、“000000”Hである。最大値レジスタ33は、乱数の値の最大値を設定する。デフォルトの値は、“FFFFFF”Hである。

【0069】発生個数レジスタ34は、発生された乱数の値の個数を設定する。この発生個数レジスタ34の値は有効値加算回路38の結果と比較され、有効値加算回路38の結果値が発生個数レジスタ34の値と等しくなると、乱数発生が停止する。デフォルトの値は、“FFFFFF”Hである。

【0070】これら各レジスタ31～34は全て12ビット構成でなる。乱数発生制御回路35は、乱数発生Aフィールド28のプログラミング値によって、コマンドデコードを行ない、その結果を乱数発生・補正回路36に通知する。

【0071】乱数発生・補正回路36は、乱数を生成する乱数発生回路と初期値の補正回路とでなる。この乱数発生・補正回路36に含まれる乱数発生回路は、「2を法とする原始多項式」に基づいて生成されるM系列発生回路を採用している。高速に生成する必要性から、24段のD型フリップフロップとEX-OR回路で構成され、この回路が1ビット当たりの乱数を発生する。従って、M系列発生回路が、24本必要であり、それぞれのM系列から出力されるデータを並列に並べてビットの重みをつけることにより、その値を乱数系列値とする。また、D型フリップフロップ（F/F）とEX-OR回路でM系列発生回路を構成した場合、初期値がすべて「0」のとき、出力は常に「0」となるので固定的な手法で補正を加えている。この乱数発生回路の構成は後に詳細に説明する。

【0072】同値検出回路／記録メモリ37は、同じ値が生成されたときは値の破棄処理と再生成を制御する。回路的には、24×1ビットメモリに発生した値のメモリセルにマークを付けていく（“1”をセットする）方法をとっている。乱数モジュール起動時に24×1ビットメモリをクリアしておき、生成された値でアドレッシングされたセルを読み出し、“0”であれば、“1”を書き込み、有効値加算回路38に通知して当該加算回路のカウンタをインクリメントする。逆に、セルを読み出したときに、“1”であれば、前に同じセルがアクセスされたことを意味しており（すでに値が生成されている）、有効値加算回路38を介して、再生成の要求を乱数発生制御回路35に通知する。

12

【0073】データ・ストレージ・メモリ（発生値格納メモリ）39は、生成された乱数系列の値を順次格納していく。このメモリへの書き込み時に於けるアドレッシングは、有効値加算回路38から与えられ、読み出し時は、有効値加算回路38あるいはALPGモジュール15からアドレッシングがプログラム選択できるようになっている。このメモリ39は、24×24ビット構成の大容量メモリモジュールにより構成される。

【0074】このメモリ39に、一度、乱数系列の値を格納しておけば、ALPGモジュール15からアドレッシングして、読み出すことにより、試験デバイスのアドレスパターンやデータパターンとして利用でき、しかも非常に効率のよい方法が採れる。

【0075】出力バッファ回路41は、乱数発生・補正回路36に含まれる乱数発生回路のデータをデータセクタモジュール17に転送する。図4は、「2を法とする原始多項式」に基づいて生成される「M系列発生回路」の基本回路である。この回路は、24段のフリップフロップ（以下F/Fと称す）とEX-OR（以下XORと称す）回路で構成する。

【0076】M系列発生回路が、m段のF/FとXORで構成されると、次のような特性を持つ。

- (1) 2^{m-1} という周期を持つ。
- (2) 1周期内に、「1」が 2^{m-1} 個、「0」が $2^{m-1}-1$ 個存在する。「0」と「1」は同じ確率で表れる。
- (3) 初期値がすべて「0」の時は、「0」と「0」のXORの出力は「0」となるので、M系列の出力も常に「0」となり乱数の発生ができない。

【0077】このシステム仕様では、最大24ビット分の値が必要なので、(1)の関係より、 $m=24$ としなければならない。 $m=24$ の時の周期は、「16, 17, 217」であり、丁度「16Mの空間」になる。

【0078】また、XOR回路を採用しているので(3)の問題を解決する必要がある。 $m=24$ の時の原始多項式は、

$$H(x) = X^{24} + X^4 + X^3 + X + 1$$

であるから図4の基本回路と一致する。

【0079】図5は上記したM系列発生回路の1段当たり（但し2、4、5段を除く）の回路構成を示す図である。この回路はF/Fに初期値を設定するデータセクタ回路が追加されている。A入力、B入力のいずれかがS入力の値によって選択される。

【0080】図6は上記したM系列発生回路の2、4、5段の回路構成を示す図である。この回路は、図5のA入力にEX-OR回路を追加したものである。図5及び図6に示す左の回路を右側のブロック図に置き換えて具体的にM系列発生回路を構成した具体例を図7に示している。

【0081】図8は上記した乱数発生・補正回路36の詳細な回路構成を示す図である。図8に於いて、81は

13

初期値レジスタ31の値が転送されるレジスタ(REG)81であり、乱数発生開始時は初期値レジスタ31と同じ値がセットされている。82はM系列発生回路であり、この回路が24個配置されて、ビットの重み付け回路83が構成されている。この回路83のDOU23~DOU00が乱数値として、テストパターンになる。

【0082】84は乱数発生開始時に、レジスタ81の「0」値を検出する、オールゼロ検出回路である。このオールゼロ回路84でオールゼロが検出されると、当該回路84「1」から出力される。この出力は、M系列発生回路82の任意の入力に接続されており、必ず、M系列の乱数が生成されるようになっている。

【0083】DOU00のM系列発生回路のみ、2カ所(B22とB00)に接続されているのは、他のM系列発生と同じ系列になるのを防止するためである。1番目に発生する値を初期値レジスタの値と同値にするには、最終段のF/F(X²⁴項)にレジスタの値をそのまま転送することにより対応している。

【0084】上記したような乱数発生モジュール(高速プログラマブルパターン発生装置)を用いてパターン発生器を構成したことにより、従来のパターン発生器では不可能なテストパターン列が生成できる。しかも、乱数生成部がモジュール化されており高度なプログラミング知識を必要としない。また、ALPGと同時に起動が可能であるため、テストパターンの作成方法の手段が増えることにより、非常に効率の良いテストパターンの発生が期待できる。

【0085】また、乱数系列による、アドレスパターンとデータパターンの生成を同時に実行処理する構成としたことにより、メモリセルに、あるデータを書き込むとき、そのデータの補数をとる(裏パターン)ことにより、すべてのビットセルの試験ができる。乱数系列によるデータパターンの生成は同様の方法で、裏パターンを生成することが可能である。しかし、アドレスパターンは、この方法では、すべてのアドレス値が生成できるという保証はない。これは、すべてのメモリセルに対して、リード/ライトが行われないことにつながる(検出率が悪くなる)。上記した実施形態の乱数パターン発生器は、同値を検出して、破棄無効化処理を行うことにより、異値が発生するまで乱数値を生成する手法で、アドレスパターンとして使用可能にしている。

【0086】また、乱数発生手段に、乱数生成値の同値検出機能を備え、異なった値のみを出力可能な構成としたことにより、異なった値のみ格納すればよいことからメモリ容量を節約することができる。また、読み出して利用する場合も最小の読み出しサイクルすべての値を実現することが可能で、非常に効率のよいテストパターンが生成できる。

【0087】また、上記した実施形態に於ける乱数発生

14

手段は、初期値レジスタのデータをM系列発生回路の最終段にセットする構成としたことにより、確実に初期値を出力できる。M系列発生回路を単純に並列配置した場合は、最初の乱数生成にプログラムされた値と同じ初期値を生成することはできない。また、D型フリップ/フロップとEX-OR回路で構成したM系列発生回路は、すべて0の場合、乱数を生成することができないが、上記実施形態によるM系列発生回路は固定的手法であるが、オールゼロ検出回路を設けてこの不具合を回避している。

【0088】上記した本発明の実施形態によるパターン発生器、または当該パターン発生器を用いたテストシステムは、パーソナルコンピュータの拡張ボードとして容易に提供でき、パーソナルコンピュータの拡張スロットに挿入して用いることにより、容易にかつ安価に、例えば簡易パターン発生装置、ICテストシステム等が実現できる。また、上記実施形態による乱数発生モジュールは、ノイズ・ジェネレータ、その他種々の計測、通信、符号化などの信号源としても利用可能である。

【0089】

【発明の効果】以上詳記したように本発明によれば、従来の装置を有効に活用して、高度なプログラミング知識を一切必要とせずに、簡単かつ安価な構成で、従来のパターン発生器では達成し得ない検出率の高いランダムなテストパターン列が生成でき、効率の良いテストパターンの発生が期待できるパターン発生器が提供できる。

【図面の簡単な説明】

【図1】本発明の実施形態によるパターン発生器の構成を示すブロック図。

【図2】上記実施形態に於けるマイクロプログラムメモリの1ワード分のフォーマットを示す図。

【図3】上記実施形態に於ける乱数発生モジュールの1台分の構成を示すブロック図。

【図4】上記実施形態に於ける乱数発生モジュールに於ける「2を法とする原始多項式」に基づいて生成されるM系列発生回路の基本回路構成を示す図。

【図5】上記図4に示すM系列発生回路の1段当たり(但し2、4、5段を除く)の回路構成を示す図。

【図6】上記図4に示すM系列発生回路の1段当たり(2、4、5段)の回路構成を示す図。

【図7】上記図5及び図6に示す回路をもとに構成された本発明の実施形態によるM系列発生回路の具体的な回路構成を示す図。

【図8】上記実施形態に於ける乱数発生・補正回路の詳細な回路構成を示す図。

【図9】代表的なアドレス関数によるパターン発生数の計算式とサイクルタイムが100nsのランダムアクセスメモリを想定したときのテスト時間を比較した例を示す図。

【図10】ディジタル半導体テストシステムの代表的な

15

構成例を示すブロック図。

【図11】従来のパターン発生器の構成を示すブロック図。

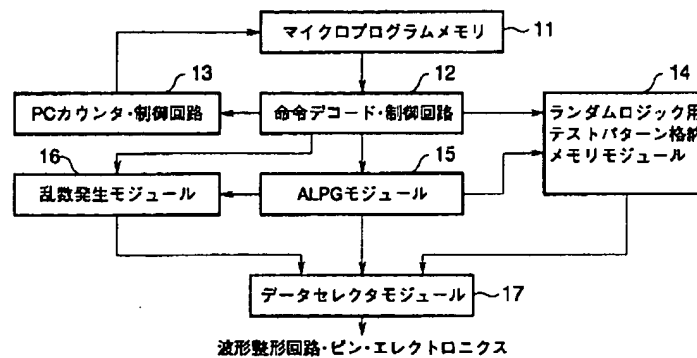
【符号の説明】

- 11…マイクロプログラムメモリ
 12…命令デコード・制御回路
 13…PCカウンタ・制御回路
 14…メモリモジュール
 15…アルゴリズムパターン発生器 (ALPGモジュール)
 16…乱数発生モジュール
 17…データセクタモジュール
 21…命令フィールド
 22…オペランドフィールド
 23…I/O制御フィールド
 24…コンパレータ制御フィールド
 25…RTTC (Real-Time Clock Timing Control) 関連制御フィールド
 26…ALPG制御フィールド

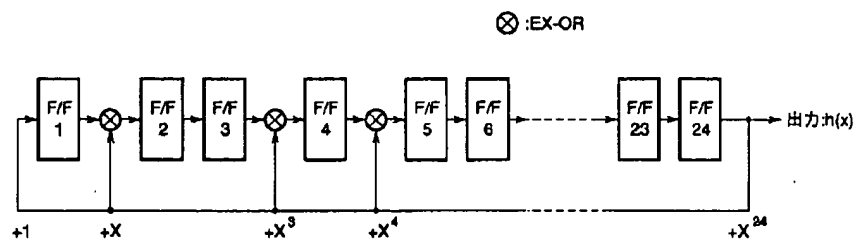
16

- 27…乱数発生制御フィールド
 31…乱数系列の初期値 (シード) を設定するためのレジスタ
 32…乱数の値の最小値を設定するレジスタ
 33…乱数の値の最大値を設定するレジスタ
 34…発生された乱数の値の個数を設定するレジスタ
 35…乱数発生制御回路
 36…乱数発生・補正回路
 37…同値検出回路/記録メモリ
 38…有効値加算回路
 39…データ・ストレージ・メモリ (発生値格納メモリ)
 41…出力バッファ回路
 81…初期値レジスタの値が転送されるレジスタ (REG)
 82…M系列発生回路
 83…ビットの重み付け回路
 84…オールゼロ検出回路

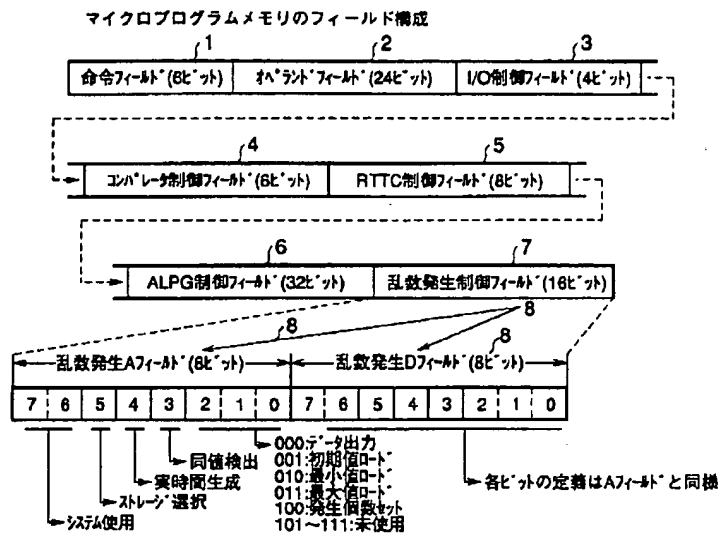
【図1】



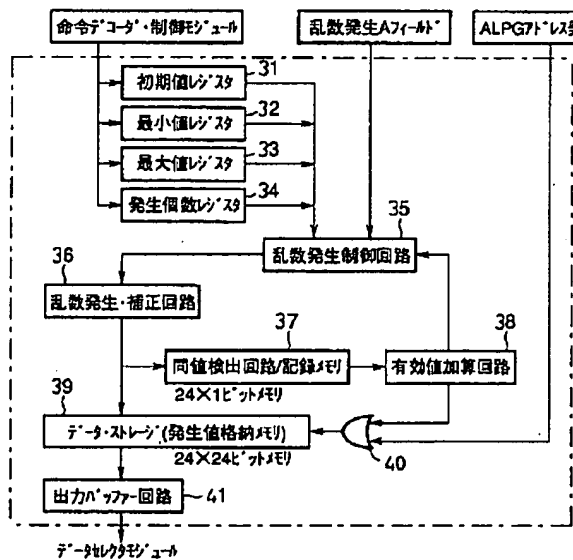
【図4】



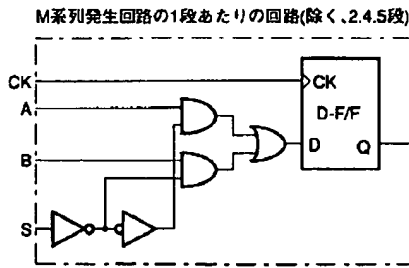
【図2】



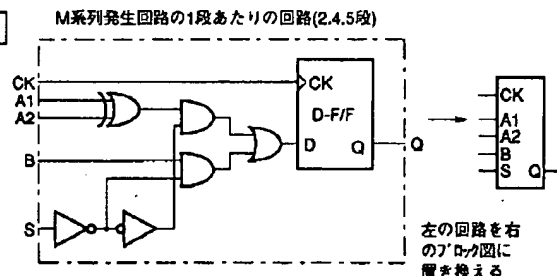
【図3】



【図5】



【図6】

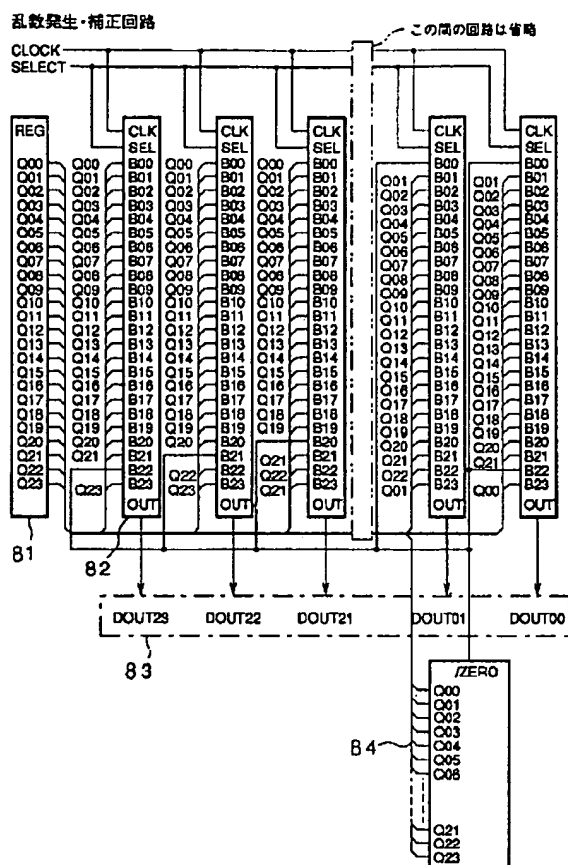
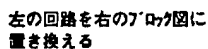


【図9】

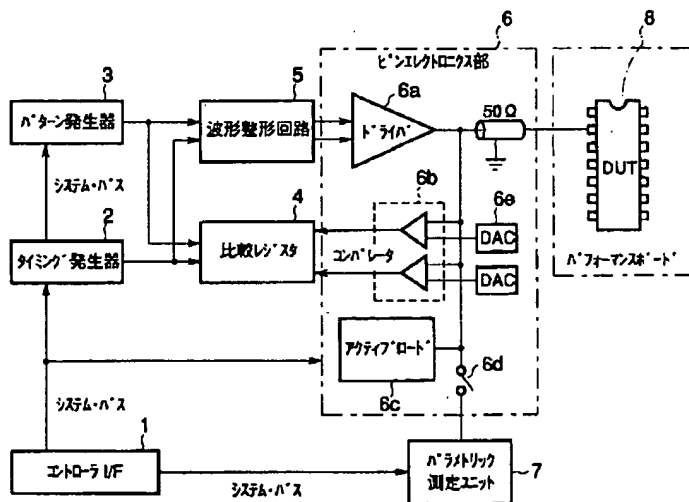
代表的なアドレス関数と試験時間

種類	パターン名	発生数	1M	4M	16M
N	MemoryScan	2N x2	0.4秒	1.7秒	6.7秒
	Marching	5N x2	1.0秒	4.2秒	16.8秒
N ^{3/2}	X-Pingpong	2N ^{3/2} x2	7.2分	57分	7.6時間
	X-Galloping	3N ^{3/2} x2	10.7分	1.4時間	11.5時間
N ²	Pingpong	2N ² x2	122時間	81日	1303日
	Galloping	3N ² x2	182時間	122日	1955日

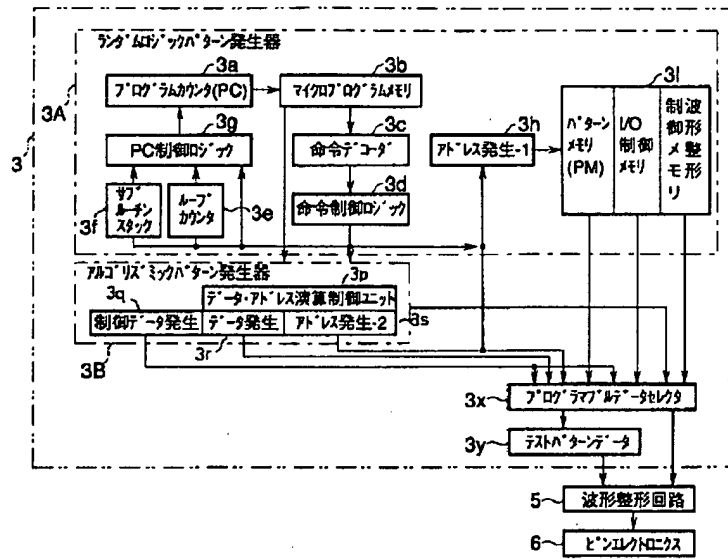
【図8】



【図10】



【図11】



PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-184678

(43)Date of publication of application : 09.07.1999

(51)Int.Cl.

G06F 7/58

(21)Application number : 09-358015

(71)Applicant : TOSHIBA CORP

(22)Date of filing : 25.12.1997

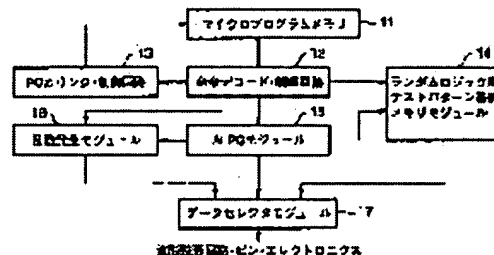
(72)Inventor : HAMAZAKI TOSHIHARU

(54) PATTERN GENERATOR

(57)Abstract:

PROBLEM TO BE SOLVED: To provide a pattern generator capable of generating random test pattern streams with a high detection rate, which can not be achieved by a conventional pattern detector, without requiring any high-level programming knowledge while effectively utilizing a conventional device without damaging conventional functions by additionally providing the conventional pattern generator with a random number generation module (high-speed programmable pattern generator).

SOLUTION: The pattern generator for generating a test pattern for IC test check or the like according to the operation sequence of a program stored in a microprogram memory is additionally provided with a random number generation module 16 for generating programmable patterns at a high speed under the control of the above program and with a memory module 14 for storing the test pattern data for random logic generated by the relevant module and further, simultaneous use with an algorithmic pattern generator(ALPG) module 15 is enabled.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of

* NOTICES *

JPO and NCIPi are not responsible for any damages caused by the use of this translation.

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.**** shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

CLAIMS

[Claim(s)]

[Claim 1] The pattern generator characterized by coming to provide the random-number-generation module which generates a programmable test pattern under the above-mentioned program control, and the memory module which stores the test pattern data generated by said random-number-generation module in the pattern generator which generates the random test pattern of trial checking of semiconductor memory etc. by the operating sequence according to the program stored in micro program memory.

[Claim 2] The pattern generator according to claim 1 which increases the number of decodings of an instruction decoder and enabled it to generate the random test pattern of trial checking of semiconductor memory etc. in the pattern generator equipped with an algorithmic pattern generating means to generate a test pattern by the operating sequence according to the program stored in micro program memory while adding the control field of a random-number module to micro program memory.

[Claim 3] A random-number-generation module is a pattern generator according to claim 1 or 2 which is the count of pattern generating of "N pattern", and generates the random address and data.

[Claim 4] A random-number-generation module is the pattern generator of claim 1, 2, or 3 publications with the processing means by the random-number sequence which carries out executive operation of the generation of an address pattern and a data pattern to coincidence.

[Claim 5] Generating of the random-number sequence by the random-number-generation module is a pattern generator according to claim 4 generated automatically by the program control by the side of a test system being possible, and programming initial value.

[Claim 6] The pattern generator according to claim 1, 2, 3, or 4 which was equipped with the equivalent detection function of a random-number generation value, and enabled the output only of a different value.

[Claim 7] A random-number-generation module is claim 1, 2, 3, 4, or 6 written pattern generator with the function which skips the same value until it becomes the value from which the random number of the address value generated and a data value differs.

[Claim 8] A random-number-generation module is claim 1 which comes to have the function which generates a random-number value automatically after constituting a multi-bit output circuit, using an M sequence generating circuit two or more and outputting initial value, 2, 3, 4, 6, or 7 written pattern generator.

[Claim 9] Claim 1 which comes to have the function to generate a test pattern on real time, and the function which stores the value generated beforehand and uses the value as a test pattern, 2, 3, 4, 6, or 7 written pattern generator.

[Claim 10] Claim 1, 2, 3, 4, 6, or 7 written pattern generator mounted in the board for an escape inserted in the expansion slot of a personal computer.

[Translation done.]

* NOTICES *

JPO and NCIPi are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[Field of the Invention] This invention relates to the pattern generator applied to various electronic equipment and components, such as for example, a semi-conductor test system, a board test system, modular component trial test equipment, and a measuring machine machine.

[0002] Moreover, this invention applies a flash memory etc. to the test system of the semiconductor memory equipment in which rapid access is possible with large capacity, and relates to a suitable pattern generator. Moreover, this invention relates to the pattern generator which can be mounted in the add-in board inserted in expansion slots, such as a personal computer.

[0003]

[Description of the Prior Art] The information impressed to DUT (Device Under Test) to examine and the direction of the signal impressed to the expected value and DUT which are outputted from DUT are beforehand stored as a test pattern, respectively, and the pattern generator applied to test inspection of a semi-conductor test system, a board test system, etc. is used in case the functional test of DUT is mainly performed.

[0004] The test pattern which can be treated with this pattern generator can be divided roughly into the thing for the random logic IC, such as general digital IC and CPU, and the thing for Memory IC which can generate an address function and a data function efficiently. The former is called a random logic pattern generator and the latter is called an algorithmic pattern generator. Furthermore, the pattern generator which carried these two functions also exists. Both common feature is a point that it is controlled by the microprogram control system and a flexible test pattern can be generated now.

[0005] Drawing 10 shows the typical example of a configuration of a digital semi-conductor test system. In the test system shown in this drawing 10, the function of a pattern generator 3 is as above-mentioned. A timing generator 2 is a module which defines time information, and each information on the timing generator 2 concerned and a pattern generator 3 is impressed to a waveform shaping circuit 5.

[0006] A waveform shaping circuit 5 is a circuit which makes the wave format impressed to DUT based on the information on a pattern generator 3 and a timing generator 2. As a class of wave format, there are RZ (Return to Zero), NRZ (Non Return to Zero), EX-OR (Exclusive OR), etc.

[0007] After the wave carried out by the above-mentioned waveform shaping circuit 5 at the time of plastic surgery and ** is inputted into driver 6a of pin electronics and clamped by the programmed high/low level, it is impressed to DUT8 as an input wave through the signal line of a fixed impedance (50ohms).

[0008] On the other hand, after the output from DUT8 is compared with the reference value (threshold level) of DAC6e by analog comparator 6b in an output level, it is transmitted to the comparison register 4.

[0009] In the comparison register 4, the expectation value pattern inputted from the hour entry (strobe) and pattern generator 3 of a timing generator 2 is compared, and functional success or failure is judged.

[0010] The parametric measurement unit 7 is a module which measures an electrical potential difference

and a current. Relay 6d which branches a signal is prepared in the pin electronics section 6. The program control of each functional module containing the above-mentioned pattern generator 3 is altogether carried out through the controller interface 1 and a system bus 9.

[0011] Drawing 11 is the block diagram showing the configuration of the conventional pattern generator. Here, the configuration including the both sides of the pattern generator for random logic (random logic pattern generator) and the pattern generator for memory (algorithmic pattern generator) is shown.

[0012] Generally as for the pattern generator of a configuration as shown in this drawing 11, random logic pattern generator 3A controls the whole equipment. In drawing 11, 3a thru/or 3i make the component of random logic pattern generator 3A, and 3a is a program counter (PC) and is a counter for accessing micro program memory 3b. Micro program memory 3b stores the instruction which determines the operating sequence of a pattern generator 3, and the data of data address operation control unit 3p of algorithmic pattern generator 3B, and control data generating section 3q. The target test pattern can be generated by storing various instructions in this micro program memory 3b, and programming an operating sequence.

[0013] Instruction decoder 3c decodes the data stored in the instruction field of micro program memory 3b, and chooses an instruction control logic 3d circuit. The classes of instruction of a pattern generator 3 are some kinds, and 40 or less cases of the number of instructions are almost the case.

[0014] This means many [that instruction processing must be carried out more to a high speed, and / if only the sequence of pattern generating is observed / functionally]. Loop counter 3e and subroutine stack 3f are the registers for making generating of the same pattern generate without futility (efficiently). these registers -- usually -- 2 -- or it has about 16 pieces and a multiplex loop formation and multiplex nesting are possible.

[0015] PC control logic 3g, the instruction executed next is controlled and the value is transmitted to program counter 3a. Moreover, program counter 3a is controlled based on flags, such as starting (start) of a pattern generator 3, a halt (stop), a halt (pause), a timer interrupt, and match interruption.

[0016] 3h of address generation circuits is a circuit for generating the address which accesses memory group 3i. Memory group 3i is a bulk memory group which memorizes each information, such as a test pattern impressed to a trial device (DUT8), and I/O-hardware-control information on pin electronics, waveform-shaping information (wave format). The output data from such memory turn into data (test pattern data) actually impressed to a trial device (DUT8).

[0017] The above is the configuration and function of random logic pattern generator 3A. On the other hand, algorithmic pattern generator (ALPG module is called below) 3B opts for actuation based on the information on the ALPG control field of micro program memory 3b, and the information inputted from instruction control logic 3d.

[0018] ALPG module 3B is the pattern generator mainly realized for the purpose of generating of test patterns, such as Memory IC, it is simple for the approach of outputting the result which carried out data processing, and since it does not need the memory which stores a test data, it is realizable by low cost.

[0019] Data address operation control unit 3p of ALPG module 3B is a circuit which calculates the data pattern impressed to a memory device, and an address pattern. The information which determines operation expression is given from the ALPG control field of micro program memory 3b, and information, such as a count of a repeat of an operation, maximum of the result of an operation, and the minimum value, is given from instruction control logic 3d.

[0020] Control data generating section 3q of ALPG module 3B performs the light / lead signal control, and I/O hardware control to trial memory. Moreover, the control which makes it accessible also performs address information generated in 3s of address generation -2 circuits for memory group 3i.

[0021] The above is the configuration and function of algorithmic pattern generator (ALPG) 3B. As described above, the pattern generator carried in the conventional test system has many places followed to a programming technique, and can create a complicated test pattern depending on the programming approach.

[0022]

[Problem(s) to be Solved by the Invention] However, in the above-mentioned pattern generator, while programming an input test pattern to a trial device, it is necessary to program the response pattern from a trial device, i.e., an expectation value pattern, as a test pattern to coincidence. Therefore, when the force of a logic simulation was not borrowed to test patterns, such as random logic LSI like a microprocessor, creation of a test pattern was difficult. Moreover, generating of high-speed test patterns, such as a memory device from which the access time of read/write, such as a flash memory, differs, for example, is also received. If a means to memorize all that high-speed I / O data tends to be needed and it is going to realize the test facility by two or more sorts (N pattern) of random pattern generating to the test of this seed high-speed memory device especially with a big capacity The memory apparatus of a huge capacity in which rapid access is possible was needed, and since the system configuration became it is remarkable and complicated, the problem of being scarce was in implementability.

[0023] This invention was made in view of the above-mentioned actual condition, utilizes conventional equipment effectively, without needing no advanced programming knowledge, can generate a random test pattern train with the detection ratio are an easy and cheap configuration and high which cannot be attained in the conventional pattern generator, and aims at offering the pattern generator which can expect generating of an efficient test pattern.

[0024]

[Means for Solving the Problem] This invention enabled it to generate an impossible test pattern train with the conventional pattern generator roughly by adding a random-number-generation module to the pattern generator of a function conventionally. And advanced programming knowledge is not needed from the modularization of the random-number generation section being carried out. Moreover, since ALPG and coincidence starting are possible, creation of a test pattern increases and generating of a very efficient test pattern can be expected.

[0025] The pattern generator carried in the test system of the above-mentioned present condition has many places followed to a programming technique, and can create a complicated test pattern depending on the programming approach. However, while programming an input test pattern to a trial device, it is necessary to program the response pattern from a trial device, i.e., an expectation value pattern, as a test pattern to coincidence.

[0026] If the force of a logic simulation is not borrowed, creation of a test pattern is in the difficult condition, but if another view is carried out, I hear that test patterns, such as random logic LSI like a microprocessor, have the creation approach of a test pattern, and a tool for creating, and there are.

[0027] This invention is made not for the test pattern creation for random logic LSI but for the test pattern generation for Memory IC. Generally, the test pattern for memory controlled the arithmetic circuit of ALPG how, and is applied to whether a test pattern is generated. As a class of test pattern generation, they are only generation of the address function which accesses trial memory, generation of the data function which carries out read/write of the memory cell, and a light / change pattern to lead. Among these, processing how etc. will be programmed when reaching whether a result-of-an-operation value is initialized or a complement will be taken if it has generated according to some easy sequence-of-numbers procedures -- whether what it carries out whether it adds or it subtracts or by what value it breaks -- and an address function becomes what kind of value, maximum, or the minimum value.

[0028] On the other hand, the technique of generating a data pattern by giving initial value generally and repeating the complement of the value, although the test pattern function is possible also for the same operation as an address function is taken. But although the address function may take out the address value which accesses memory, a data pattern needs to perform the operation (programming) which can generate the same value as the data value when writing in, if the address of memory is determined on the function in comparison with expected value at the time of a lead.

[0029] When the class of address function of memory is roughly divided, there are "N pattern", an "N³/square pattern", an "N square pattern", etc. (however, when N is made into the addressing number of bits). Other than "N pattern", on the relation of test time amount, although a detection ratio is very high, in the latest bulk memory, it is not practical. Therefore, it is [how] efficient by "N pattern", and programming of an address function and data function generation is required whether to gather a

detection ratio moreover.

[0030] The example which compared test time amount when the formula and the cycle time of pattern occurrences by the typical address function in this case assume the random access memory for 100ns is shown in drawing 9.

[0031] This invention is the count of pattern generating of "N pattern", and aims at offering the module which generates the random address and data. Originally this module originates in that access of memory is random access, and it being data in which the data written also had a random value.

[0032]

[Embodiment of the Invention] This invention is realized by adding a "random-number-generation module" and "the memory module which stores the generated data" to the conventional pattern generator, as mentioned above. It decided to satisfy the following specification from the field of application in development of this pattern generator.

[0033] (1) The function of the pattern generator of . former shall not be made into a sacrifice.

It realizes by adding the module for random number generation to the conventional functional module of a pattern generator. Control of this module corresponds by increasing the number of decodings of an instruction decoder while adding the control field of a random-number module to micro program memory.

[0034] (2) Generate automatically by programming initial value (only giving a seed to a random number) without performing that the program control by the side of a test system is possible for generating of . random-number sequence, and programming of the random number generation itself.

[0035] In order to simplify programming, it is not necessary to carry out user programming about generating of the random number sequence itself, and considers as the method which generates different random number sequence automatically by giving only initial value.

[0036] (3) . address and the random number of a data value should skip the same value (it cancels), and should have the function to make a new value generate.

If the value of the usual random number is adopted as it is, the same value will surely occur. For example, when this value is used for addressing of memory, the same memory cell will be accessed and it cannot be said as efficient access. Moreover, there will also be no guarantee that all memory cells are accessed once [at least]. Therefore, the function to skip is realized until a different value is generated.

[0037] (4) The generating approach of the value of . random number stores the function generated on real time (real time), and the value generated beforehand, and have the function which uses the value as a test pattern.

[0038] Although a circuit becomes easy as a test pattern, without storing the value temporarily since it can be impressed by the trial device when generating a random-number sequence on real time, there is no guarantee as which all memory cells are chosen. Although what is necessary is just to skip until the value from which the same value differed is generated, in the pattern generator which must perform one test pattern in 100ns, it is impossible in time. Therefore, in the function to generate a pattern on real time, it considers only as the value of the data pattern of memory, and assumes not using addressing for the purpose.

[0039] The memory to store is needed while activation of the useless test pattern which is not can be performed, since the approach of storing the value of a random number beforehand before activation initiation of pattern generating can be skipped until a different value is generated in the case of the same value. This function can be used also for addressing of trial memory.

[0040] With reference to a drawing, 1 operation gestalt of this invention is explained below. Drawing 1 is the block diagram showing the configuration of the pattern generator by the operation gestalt of this invention. The pattern generator shown in this drawing 1 is inserted in expansion slots, such as a personal computer, and is used, for example, can be offered as various add-in boards for trial test equipment, such as a semi-conductor test system and a board modular component test system.

[0041] In drawing 1, 11 is micro program memory which stores the control program of the pattern generator performed by CPU which is not illustrated etc., and 12 is instruction decoding and a control circuit which decodes the micro program stored in the micro program memory 11, and generates a

control signal.

[0042] 13 is PC counter and a control circuit which carries out updating control of the contents of the micro program memory 11. 14 is a memory module which stores the test pattern for random logic generated with the algorithmic pattern generator 15 mentioned later. 15 is an algorithmic pattern generator with which actuation of test pattern data generation of Memory IC etc. is determined as the bottom of micro program control (on basis [information / on the ALPG control field of the micro program memory 11, and control field]), and calls an ALPG module here. 16 is the target random-number-generation module in this invention which enables generating of the test pattern with the sufficient effectiveness which continues variably which was not able to be generated in the former, and the concrete configuration and concrete function are mentioned later. 17 is a data selector module which chooses and outputs the above-mentioned memory module 14, the algorithmic pattern generator 15, and the test pattern generated in the random-number-generation module 16 grade.

[0043] In the pattern generator of a configuration of being shown in above-mentioned drawing 1, the micro program memory 11 stores the program which controls a pattern generator. This program memory 11 considers as 1 K word x96 bit pattern here, and is performing the bit assignment like drawing 2.

[0044] Instruction decoding and a control circuit 12 control the sequence of a pattern generator. PC counter and a control circuit 13 opt for the next actuation by the result by which instruction decoding was carried out in instruction decoding and a control circuit 12, and accesses the micro program memory 11.

[0045] A memory module 14 stores the test pattern data for random logic. Here, it is considering as the memory configuration of the number of 64 K word x circuit tester pins. The algorithmic pattern generator 15 generates the test pattern for Memory IC. Here, an ALPG module is called. This ALPG module 15 has the address (12 bits) generation arithmetic unit of X and Y here, and corresponds to the address space of 16M. Moreover, control signals, such as generation to a trial memory device and read/write, can also be generated now.

[0046] The address information generated by the above-mentioned ALPG module 15 can also carry out direct access of the data storage memory carried in the memory module 14 or the random-number-generation module 16.

[0047] The random-number-generation module 16 forms the main components of this invention, and the configuration of the interior explains it to a detail with reference to drawing 5 later. The data selector module 17 is controlled by the value of the random-number-generation control field of the micro program memory 11, and chooses and outputs a memory module 14, the algorithmic pattern generator 15, and the test pattern generated in the random-number-generation module 16 grade.

[0048] Drawing 2 shows the 1-word configuration of the above-mentioned micro program memory 11. Here, the memory configuration is made into 1K word x96 bit. Among drawing, 21 are an instruction field and are using 6 bits. Therefore, the class of instruction can be made to a maximum of 64 kinds.

[0049] 22 is an operand field and is using 24 bits. The purpose of using this field is for describing the value set as the register of each module, or setting up the branching place address in branch instruction.

[0050] 23 is the I/O-hardware-control field and is the field which specifies the address of the driver enabling memory which enables / disables the driver of pin electronics. 4 bits is used and it enables it to choose 16 kinds of driver enabling information here.

[0051] 24 is comparator control field and is the field which describes the address information which accesses the comparator enabling memory which decides whether to compare the value outputted from a trial device (DUT) with the expectation value pattern outputted from a pattern generator. 6 bits is used and it enables it to choose 64 kinds of comparator enabling information here.

[0052] 25 is RTTC (real-time timing control) related control field, and 8 bits is used here and it is mainly used for the control for a timing generator.

[0053] 26 is the ALPG control field for controlling the ALPG module 15. The ALPG module 15 of the system in this operation gestalt could generate X and the Y two-dimensional address, and is come. Since both arithmetic unit and related register are 12 bits, the trial of the memory device which has an address space to maximum 16M is possible for them.

[0054] 27 is the random-number-generation control field for controlling the random-number-generation module 16. Here, 16 bits is used, and as shown in control field 28, two random-number-generation circuits which completely have the same function are carried. This is because generating of the random-number sequence the object for the addresses and for data is enabled at coincidence.

[0055] Among these, it assumes using the random-number-generation A field in order to generate an address pattern, and using the random-number-generation D field, since a data pattern is generated. However, it is also possible to use the random-number-generation A field for a data pattern. In this case, the method of generating an address pattern from the ALPG module 15 is taken. That is, as for this pattern generator, it is possible to also make coincidence perform the ALPG module 15 and the random-number-generation module 16.

[0056] If control field 28 are explained to a detail, the low order triplet consists of a command which sets a value as each register of a random-number module, and a command which outputs the generated value among the 8-bit program fields.

[0057] The class of command in this case sets initial value as a "0" ("000") -> data output "1" ("001") -> initial value register (drawing 3 sign 31 reference). The value of the initial value to set up describes a value to set it as the above-mentioned operand field 22. The value of this register becomes the kind (seed) which determines a random-number sequence.

[0058] The minimum value of the value generated in "2" ("010") -> minimum value register (drawing 3 sign 32 reference) is set up. The setting approach is the same as that of the above.

The maximum of the value generated in "3" ("011") -> maximum register (drawing 3 sign 33 reference) is set up.

[0059] That is, the range of the value of a random-number sequence turns into the range of the value of the value <= generating value <= maximum register of a minimum value register.

The number of a random number is set as "4" ("100") -> generating number register (drawing 3 sign 34 reference).

[0060] Each above-mentioned register is 24 bit patterns, respectively, and, numerically, becomes the range to "0" - "16,777,215." However, since the memory device to examine is carrying out various bit patterns, the register of such a bit pattern is needed.

[0061] It is a "5" - "7" ("101" - "111") -> intact command (for an escape).

The triplet eye of control field 28 is a change bit for enabling / disabling an equivalent detector (drawing 3 R>3 sign 37 reference).

[0062] If an equivalent detector (drawing 3 sign 37 reference) is enabled, when the once same value as the generated value occurs, skip (it cancels and is made invalid) processing is performed, and it will repeat until the value which has not been generated once is generated. However, the more generation of random number sequence progresses, the more, this approach is also that a different probability from the value generated in the past decreases gradually, and when the worst, an ungenerable phenomenon may always generate it. If the timer interrupt is started and it does not end in fixed time amount when performing such processing, to program so that generating of a pattern may be stopped is also required.

[0063] The 4th bit of control field 28 is a control bit which specifies making effluence the generated value (pattern), without storing in data storage memory (drawing 3 sign 39 reference). When this bit is effective, an equivalent detector (drawing 3 sign 37 reference) is disabled compulsorily. That is, programming of a triplet eye is disregarded.

[0064] The 5th bit of control field 28 is a bit for changing addressing of the data storage memory 39 which stores a random number. Usually, although addressing is automatically carried out within the random-number-generation module 16, addressing from the ALPG module 15 is made possible by setting this bit.

[0065] For system usage, 2 bits of high orders of control field 28 are disregarded, even if it carries out user programming. The configuration for one set of the above-mentioned random-number-generation module 16 is shown in drawing 6.

[0066] In drawing 6, the initial value register 31 sets up the initial value (seed) of a random-number sequence. The value of this register itself is transmitted from the operand field 22 of the micro program

memory 11.

[0067] This value is processed with the principle (the fixed technique) decided in random number generation and the amendment circuit 36, and amendment is added so that a random number may be generated certainly. A default value is "FFFFFF"H (hexadecimal).

[0068] The minimum value register 32 sets up the minimum value of the value of a random number. A default value is "000000"H. The maximum register 33 sets up the maximum of the value of a random number. A default value is "FFFFFF"H.

[0069] The generating number register 34 sets up the number of the value of the generated random number. If it is compared with the result of the effective value adder circuit 38 and a value becomes equal to the value of the generating number register 34 as a result of the effective value adder circuit 38, random number generation will stop the value of this generating number register 34. A default value is "FFFFFF"H.

[0070] Each [these] registers 31-34 become by 12 bit patterns altogether. With the programming value of the random-number-generation A field 28, the random-number-generation control circuit 35 performs command decoding, and notifies the result to random number generation and the amendment circuit 36.

[0071] Random number generation and the amendment circuit 36 become in the random-number-generation circuit and the amendment circuit of initial value which generate a random number. The M sequence generating circuit generated based on "the primitive polynomial which makes 2 law" is used for the random-number-generation circuit included in this random number generation and amendment circuit 36. From the need of generating at a high speed, it consists of 24 steps of D type flip-flops, and an EX-OR circuit, and this circuit generates the random number per bit from it. Therefore, 24 M sequence generating circuits are required, and make the value a random-number sequence value by arranging in juxtaposition the data outputted from each M sequence, and giving the weight of a bit. Moreover, when an M sequence generating circuit is constituted from a D type flip-flop (F/F) and an EX-OR circuit and all initial value is "0", since an output is always set to "0", amendment is added by fixed technique. The configuration of this random-number-generation circuit is later explained to a detail.

[0072] An equivalent detector / record memory 37 controls cancellation processing and regeneration of a value, when the same value is generated. The approach ("1" is set) of attaching the mark to the memory cell of the value generated in 24x1-bit memory in circuit is taken. 24x1-bit memory is cleared at the time of random-number module starting, the cel by which addressing was carried out with the generated value is read, if it is "0", "1" will be written in, it will notify to the effective value adder circuit 38, and the counter of the adder circuit concerned will be incremented. On the contrary, if it is "1" when a cel is read, it will mean that the cel same in front was accessed (the value is already generated), and the demand of regeneration will be notified to the random-number-generation control circuit 35 through the effective value adder circuit 38.

[0073] The data storage memory (generating value storing memory) 39 carries out sequential storing of the value of the generated random-number sequence. Addressing at the time of the writing to this memory is given from the effective value adder circuit 38, and it has come to be able to carry out the program selection of the addressing from the effective value adder circuit 38 or the ALPG module 15 at the time of read-out. This memory 39 is constituted by the mass memory module of 24x24 bit pattern.

[0074] By carrying out addressing and reading from the ALPG module 15 to it, once it stores the value of a random-number sequence in this memory 39, it can use as the address pattern and data pattern of a trial device, and an approach efficient moreover very can be taken.

[0075] The output-buffer circuit 41 transmits the data of the random-number-generation circuit included in random number generation and the amendment circuit 36 to the data selector module 17. Drawing 4 is the basic circuit of the "M sequence generating circuit" generated based on "the primitive polynomial which makes 2 law." This circuit consists of 24 steps of flip-flops (F/F is called below), and an EX-OR (Following XOR is called) circuit.

[0076] When an M sequence generating circuit is ***** (ed) by m steps of F/F, and XOR, it has the following properties.

(1) $2m-1$ It has the period to say.

(2) In 1 period, "1" is $2m-1$. An individual and one to one $2m - "0"$ exist. "0" and "1" appear in the same probability.

(3) Since the output of XOR of "0" and "0" is set to "0" when all initial value is "0", the output of an M sequence is also always set to "0", and cannot perform generating of a random number.

[0077] In this systems specification, since the value for a maximum of 24 bits is the need, it must be referred to as $m=24$ from the relation of (1). The period at the time of $m=24$ is "16,177,217", and becomes "space of 16M" exactly.

[0078] Moreover, since the XOR circuit is adopted, it is necessary to solve the problem of (3). Since the primitive polynomial at the time of $m=24$ is $H(x) = X^{24} + X^4 + X^3 + X + 1$, it is in agreement with the basic circuit of drawing 4.

[0079] Drawing 5 is drawing showing the circuitry per step (however, 2 and 4 or 5 steps are removed) of the above-mentioned M sequence generating circuit. The data selector circuit where this circuit sets initial value as F/F is added. A input or B input is chosen by the value of S input.

[0080] Drawing 6 is drawing showing 2 of the above-mentioned M sequence generating circuit, and 4 or 5 steps of circuitry. This circuit adds EX-OR circuit to A input of drawing 5. The example which transposed the circuit of the left shown in drawing 5 and drawing 6 to the right-hand side block diagram, and constituted the M sequence generating circuit concretely is shown in drawing 7.

[0081] Drawing 8 is drawing showing the detailed circuitry of the above-mentioned random number generation and amendment circuit 36. In drawing 8, 81 is the register (REG) 81 with which the value of the initial value register 31 is transmitted, and the same value as the initial value register 31 is set at the time of initiation of random number generation. 82 is an M sequence generating circuit, these 24 circuits are arranged and the weighting circuit 83 of a bit is constituted. DOUT23-DOUT00 of this circuit 83 become a test pattern as a random-number value.

[0082] 84 is an all zero detector which detects "0" values of a register 81 at the time of random-number-generation initiation. If all zero are detected in this all zero circuit 84, it will be outputted from the circuit 84 concerned "1." This output is connected to the input of the arbitration of the M sequence generating circuit 82, and the random number of an M sequence is surely generated.

[0083] Only the M sequence generating circuit of DOUT00 is connected to two places (B22 and B00) for preventing becoming the same sequence as other M sequence generating. In order to make into the value and the equivalent of an initial value register the value generated in the 1st, it corresponds by transmitting the value of a register to F/F (X^{24} term) of the last stage as it is.

[0084] By having constituted the pattern generator using a random-number-generation module (high-speed programmable pattern generator) which was described above, an impossible test pattern train is generable with the conventional pattern generator. And the modularization of the random-number generation section is carried out, and it does not need advanced programming knowledge. Moreover, since ALPG and coincidence starting are possible, when the means of the creation approach of a test pattern increases, generating of a very efficient test pattern is expectable.

[0085] Moreover, when writing a certain data in a memory cell by having considered as the configuration which carries out executive operation of the generation of the address pattern by the random-number sequence, and a data pattern to coincidence, the trial of all bit cels can be performed by what the complement of the data is taken for (flesh-side pattern). Generation of the data pattern by the random-number sequence is the same approach, and it is possible to generate a flesh-side pattern. However, there is no guarantee that an address pattern can generate all address values by this approach. This leads to read/write not being performed to all memory cells (a detection ratio worsens). By detecting the equivalent and performing cancellation nullification processing, the random-number pattern generator of the above-mentioned operation gestalt is the technique of generating a random-number value until a different value occurs, and it is made usable as an address pattern.

[0086] Moreover, since what is necessary is to store in a random-number-generation means only a different value by having had the equivalent detection function of a random-number generation value, and having considered only a different value as the configuration in which an output is possible,

memory space can be saved. Moreover, also when beginning to read and using, it is possible to realize the values of all the minimum read-out cycles, and a very efficient test pattern can be generated.

[0087] Moreover, the random-number-generation means in the above-mentioned operation gestalt can output initial value certainly by having considered as the configuration which sets the data of an initial value register to the last stage of an M sequence generating circuit. When the parallel arrangement of the M sequence generating circuit is carried out simply, the same initial value as the value programmed by the first random-number generation cannot be generated. Moreover, in the case of 0, not all the M sequence generating circuits constituted from a D mold flip / FUIOPPU, and an EX-OR circuit can generate a random number, but although the M sequence generating circuit by the above-mentioned operation gestalt is the fixed technique, the all zero detector was prepared and this fault is avoided.

[0088] The test system using the pattern generator by the operation gestalt of above-mentioned this invention or the pattern generator concerned can be easily offered as an add-in board of a personal computer, and a simple pattern generator, IC test system, etc. can be easily realized cheaply by inserting and using for the expansion slot of a personal computer. Moreover, the random-number-generation module by the above-mentioned operation gestalt is available also as sources of a signal, such as a noise generator, other various measurement, a communication link, and coding.

[0089]

[Effect of the Invention] According to this invention, as a full account was given above, conventional equipment is utilized effectively, without needing no advanced programming knowledge, with an easy and cheap configuration, a random test pattern train with the high detection ratio which cannot be attained in the conventional pattern generator can be generated, and the pattern generator which can expect generating of an efficient test pattern can be offered.

[Translation done.]

* NOTICES *

JPO and NCIPi are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

TECHNICAL FIELD

[Field of the Invention] This invention relates to the pattern generator applied to various electronic equipment and components, such as for example, a semi-conductor test system, a board test system, modular component test equipment, and a measuring machine.

[0002] Moreover, this invention applies a flash memory etc. to the test system of the semiconductor memory equipment in which rapid access is possible with large capacity, and relates to a suitable pattern generator. Moreover, this invention relates to the pattern generator which can be mounted in the add-in board inserted in expansion slots, such as a personal computer.

[Translation done.]

* NOTICES *

JPO and NCIPi are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

PRIOR ART

[Description of the Prior Art] The information impressed to DUT (Device Under Test) to examine and the direction of the signal impressed to the expected value and DUT which are outputted from DUT are beforehand stored as a test pattern, respectively, and the pattern generator applied to test inspection of a semi-conductor test system, a board test system, etc. is used in case the functional test of DUT is mainly performed.

[0004] The test pattern which can be treated with this pattern generator can be divided roughly into the thing for the random logic IC, such as general digital IC and CPU, and the thing for Memory IC which can generate an address function and a data function efficiently. The former is called a random logic pattern generator and the latter is called an algorithmic pattern generator. Furthermore, the pattern generator which carried these two functions also exists. Both common feature is a point that it is controlled by the microprogram control system and a flexible test pattern can be generated now.

[0005] Drawing 10 shows the typical example of a configuration of a digital semi-conductor test system. In the test system shown in this drawing 10, the function of a pattern generator 3 is as above-mentioned. A timing generator 2 is a module which defines time information, and each information on the timing generator 2 concerned and a pattern generator 3 is impressed to a waveform shaping circuit 5.

[0006] A waveform shaping circuit 5 is a circuit which makes the wave format impressed to DUT based on the information on a pattern generator 3 and a timing generator 2. As a class of wave format, there are RZ (Return to Zero), NRZ (Non Return to Zero), EX-OR (Exclusive OR), etc.

[0007] After the wave carried out by the above-mentioned waveform shaping circuit 5 at the time of plastic surgery and ** is inputted into driver 6a of pin electronics and clamped by the programmed high/low level, it is impressed to DUT8 as an input wave through the signal line of a fixed impedance (50ohms).

[0008] On the other hand, after the output from DUT8 is compared with the reference value (threshold level) of DAC6e by analog comparator 6b in an output level, it is transmitted to the comparison register 4.

[0009] In the comparison register 4, the expectation value pattern inputted from the hour entry (strobe) and pattern generator 3 of a timing generator 2 is compared, and functional success or failure is judged.

[0010] The parametric measurement unit 7 is a module which measures an electrical potential difference and a current. Relay 6d which branches a signal is prepared in the pin electronics section 6. The program control of each functional module containing the above-mentioned pattern generator 3 is altogether carried out through the controller interface 1 and a system bus 9.

[0011] Drawing 11 is the block diagram showing the configuration of the conventional pattern generator. Here, the configuration including the both sides of the pattern generator for random logic (random logic pattern generator) and the pattern generator for memory (algorithmic pattern generator) is shown.

[0012] Generally as for the pattern generator of a configuration as shown in this drawing 11, random logic pattern generator 3A controls the whole equipment. In drawing 11, 3a thru/ or 3i make the component of random logic pattern generator 3A, and 3a is a program counter (PC) and is a counter for

accessing micro program memory 3b. Micro program memory 3b stores the instruction which determines the operating sequence of a pattern generator 3, and the data of data address operation control unit 3p of algorithmic pattern generator 3B, and control data generating section 3q. The target test pattern can be generated by storing various instructions in this micro program memory 3b, and programming an operating sequence.

[0013] Instruction decoder 3c decodes the data stored in the instruction field of micro program memory 3b, and chooses an instruction control logic 3d circuit. The classes of instruction of a pattern generator 3 are some kinds, and 40 or less cases of the number of instructions are almost the case.

[0014] This means many [that instruction processing must be carried out more to a high speed, and / if only the sequence of pattern generating is observed / functionally]. Loop counter 3e and subroutine stack 3f are the registers for making generating of the same pattern generate without futility (efficiently). these registers -- usually -- 2 -- or it has about 16 pieces and a multiplex loop formation and multiplex nesting are possible.

[0015] PC control logic 3g, the instruction executed next is controlled and the value is transmitted to program counter 3a. Moreover, program counter 3a is controlled based on flags, such as starting (start) of a pattern generator 3, a halt (stop), a halt (pause), a timer interrupt, and match interruption.

[0016] 3h of address generation circuits is a circuit for generating the address which accesses memory group 3i. Memory group 3i is a bulk memory group which memorizes each information, such as a test pattern impressed to a trial device (DUT8), and I/O-hardware-control information on pin electronics, waveform-shaping information (wave format). The output data from such memory turn into data (test pattern data) actually impressed to a trial device (DUT8).

[0017] The above is the configuration and function of random logic pattern generator 3A. On the other hand, algorithmic pattern generator (ALPG module is called below) 3B opts for actuation based on the information on the ALPG control field of micro program memory 3b, and the information inputted from instruction control logic 3d.

[0018] ALPG module 3B is the pattern generator mainly realized for the purpose of generating of test patterns, such as Memory IC, it is simple for the approach of outputting the result which carried out data processing, and since it does not need the memory which stores a test data, it is realizable by low cost.

[0019] Data address operation control unit 3p of ALPG module 3B is a circuit which calculates the data pattern impressed to a memory device, and an address pattern. The information which determines operation expression is given from the ALPG control field of micro program memory 3b, and information, such as a count of a repeat of an operation, maximum of the result of an operation, and the minimum value, is given from instruction control logic 3d.

[0020] Control data generating section 3q of ALPG module 3B performs the light / lead signal control, and I/O hardware control to trial memory. Moreover, the control which makes it accessible also performs address information generated in 3s of address generation -2 circuits for memory group 3i.

[0021] The above is the configuration and function of algorithmic pattern generator (ALPG) 3B. As described above, the pattern generator carried in the conventional test system has many places followed to a programming technique, and can create a complicated test pattern depending on the programming approach.

[Translation done.]